MedIO: A Program for Intelligent Clinical Data Entry

Ralph J. DeFriece, MD, MEE

Division of Medical Informatics, Duke University Medical Center, Durham NC

Several hundred million physician office visits occur in the U.S. each year. Some studies indicate that clinicians spend roughly 38% of their time documenting clinical data. Although there are many proposed benefits of using a computer system for clinical data entry, there are currently few widely accepted, uniform, yet flexible computer interfaces in use in the U.S. Until the clinical computer interface used for the transfer of clinical data approximates the speed and flexibility of the data transfer available in simple dictation and transcription systems, direct electronic capture of clinical data will be difficult.

INTERFACE DESIGN PROBLEMS

Many issues must be considered in designing a clinical interface with an efficient mechanism for transferring clinical data into the computer. A problem that greatly complicates the design of computer programs to support this transfer of data is the variability in the entire process of clinical data collection and documentation. This variability is complicated by the very nature of mental processing of clinical data by different clinicians. The term "problem space" has been used in artificial intelligence to describe the way a person internally represents the facts and rules they use to deal with a problem. The structure of a clinician's problem space for a particular clinical problem may be a very complicated structure since it involves a loosely organized framework involving a large number of facts and rules. Current approaches to designing clinical data entry programs involve trying to encode a large, complicated set of rules and data elements for each of thousands of different clinical problems and situations. Construction of computer programs in this manner is analogous to trying to extract and encode the knowledge embodied in the problem spaces of different clinicians to develop a generic clinical problem space for each of thousands of clinical problems.

DESCRIPTION OF DATA ENTRY PROGRAM

Rather than try to encode a complicated list of rules that may only apply to a limited number of clinicians, a more practical approach is to support a clinician's data collection habits by automatically organizing the data elements and rules he or she uses in a way that tries to mimic how the clinician's problem space is constructed. Two general methods for doing this involve pattern accumulation and pattern recognition.

MedIO is a C++ computer program that uses these two methods for clinical data entry of history and physical examination information. The program maintains a global pool of data elements specific to a particular class of problems (i.e. Upper Respiratory Infection for this prototype). These data elements are maintained in four lists: History Phrases, History Symptoms, Physical Exam Anatomical Site, and Physical Exam Descriptors.

A simple method of pattern accumulation is used to generate and store lists of data elements in a compressed, problem specific format. Pattern accumulation addresses the variability that occurs in the data collection process. Over several cases of a particular clinical problem, the computer program will automatically accumulate and filter a set of data elements most frequently used for that problem. Upon selection of a clinical problem from a menu, the most frequently used sets of data elements for that problem will automatically be loaded into the four list boxes.

Unlike data element collection and documentation, action patterns may not be organized on a problem specific basis. Given the large number of clinical situations a clinician might encounter, it is difficult to try to anticipate what actions a clinician might take in a variety of situations. Pattern recognition addresses this by identifying the response patterns of physicians to different clinical situations and maintaining these responses, or actions, in a fifth list box. These response patterns will automatically track how a particular clinician acts when given a particular set of data elements. An autoassociative neural network will be used to first learn, and then subsequently generate, the sets of critical data elements normally associated with a given action.

The data entry screen uses a simple, Windows multiple document interface. Upon selection of a particular clinical problem from a menu, a template document is pulled up, with an editing window and the 5 data entry list boxes mentioned previously. Data may be entered by typing in the edit window and/or by selecting data elements from the various lists of frequently used data elements and actions.

The goal of the program is to remember the most frequently used data elements and actions for a particular problem and make them quickly accessible. This design approach attempts to provide a uniform, yet flexible data collection program while also reducing the complexity and repetition inherent in the design of clinical data collection programs.